



Efficient MRF Energy Propagation for Video Segmentation via Bilateral Filters

Dr. B. R. Vikram

M.E., Ph.D., MIEEE, LMISTE, Principal of Vijay Rural Engineering College,
 NIZAMABAD (Dt.)

N. Sandeep

M.Tech, Electrical and Computer Engineering , Vijay Rural Engineering College,
 NIZAMABAD (Dt.).

Abstract—

Segmentation of an object from a video is a challenging task in multimedia applications. Depending on the application, automatic or interactive methods are desired; however, regardless of the application type, efficient computation of video object segmentation is crucial for time-critical applications; specifically, mobile and interactive applications require near real-time efficiencies. In this paper, we address the problem of video segmentation from the perspective of efficiency. We initially redefine the problem of video object segmentation as the propagation of MRF energies along the temporal domain. For this purpose, a novel and efficient method is proposed to propagate MRF energies throughout the frames via bilateral filters without using any global texture, color or shape model. Recently presented bi-exponential filter is utilized for efficiency, whereas a novel technique is also developed to dynamically solve graph-cuts for varying, non-lattice graphs in general linear filtering scenario. These improvements are experimented for both automatic and interactive video segmentation scenarios. Moreover, in addition to the efficiency, segmentation quality is also tested both quantitatively and qualitatively. Indeed, for some challenging examples,

significant time efficiency is observed without loss of segmentation quality.

Index Terms: Bi-exponential filters; bilateral filters; graph-cut; MRF; video segmentation

I. INTRODUCTION:

SEGMENTING an object of interest from an image or a video is a crucial task in many multimedia applications. Therefore, this problem has received significant interest for a long time. Moreover, both unsupervised and semi-supervised version of the problem has been attacked by many researchers. Although there exist many techniques for automatic image segmentation [1], [2], [3], the problem is considered to be illposed due to the ambiguous definition of an “object” and lack of prior information for a successful object segmentation. On the other hand; high quality, interactive image segmentation is possible due to the recent improvements. Generally, by small supervision (bounding box of the object [4], approximate boundary Manuscript received January 17, 2013; revised September 16, 2013 and January 24, 2014; accepted March 11, 2014. Date of publication March 27, 2014; date of current version July 15, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shin’ichi Satoh. O. Şener is with the Electrical and Computer Engineering Department, Cornell University, Ithaca, NY 14853 USA. This research was



conducted when he was affiliated with Middle East Technical University. K. Ugur is with Nokia Research Center, Tampere, Finland. A. A. Alatan is with the Electrical and Electronics Engineering Department, Middle East Technical University, Ankara, Turkey. Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>. Digital Object Identifier 10.1109/TMM.2014.2314069 of the object [5], a few scribbles on foreground and/or background [6], [7]), precise object boundaries or matte values can be obtained. Interactive image segmentation problem is generally converted to a two-label energy minimization problem and object boundaries are obtained via energy minimization [8] while color and texture cues of the interacted pixels are used to define this energy function [4]. Coherency of the segmentation result is also enforced via smoothness penalty term [6]. The most common energy function used in interactive image segmentation scenario is a Markov Random Field (MRF) energy [6], [4], [7], [5]. MRF energies are minimized via min-cut/ max-flow method efficiently [8]. By the introduction of the dynamic methods [9], [10] and hierarchical graph-cut [11], even near real time execution is possible under some constraints. Moreover, mobile applications are also emerging as a result of these improvements [9]. Adapting such image segmentation methods directly to the video segmentation problem is not feasible due to the additional temporal dimension, as well as the necessity to use motion information for a high quality segmentation. First of all, by the help of the motion information, it is possible to redefine the segmentation problem and solve it by an fully automatic method. However, ambiguity of the object definition still exists; object of interest might not be moving in the sequence. In order to solve this drawback, a definition of the object is revisited by the help of saliency and motion information [12], [13], [14]. Saliency-based definition of the object has already led to successful automatic video segmentation applications [15]. Although the results of this algorithm looks quite promising, both motion estimation and saliency measure are computationally expensive processes.

Therefore, it is almost impossible to apply them to any interactive or mobile multimedia application. Bottom-up processing of a video is another approach to segment videos. Bottom-up methods start with redundant over-segmentation of a video, and apply spatio-temporal clustering to obtain segmentations [16], [17]. In [16], nodes of the 3D graph is obtained via spatio-temporal over-segmentation; and, edges are added via computed optical flow. Then, clustering is performed based on the desired level of hierarchy. Using supervision for the hierarchical level selection makes this method suitable for many interactive multimedia scenarios; however, computationally expensive extraction of motion information make it impossible to apply such scenarios. In another bottom-up method [17], 2D over-segmentation of each frame is obtained. And, over-segment flows are computed via inference on a Markov chain to cluster over-segments. However, inference on Markov chains is also a computationally expensive procedure; therefore, this method is also not applicable to interactive and mobile multimedia applications. On the other hand, interactive video segmentation methods generally utilizes interactions in some keyframes [18], [19], [20]. After interaction, an energy function exploiting texture, color and interaction is defined and minimized [20], [21]. However, these approaches generate high quality segmentation results for a few frames only and require user interactions in many other frames. The main reason for this performance drawback is the lack of motion information, which is computationally quite expensive to be used. Therefore, it is desired to avoid motion information for the applications demanding efficiency. However, not only motion information, but also spatial relations are discarded in these methods due to the global models [19], [20]. It should be noted that these methods are the extensions of interactive image segmentation algorithms and most of the interactive image segmentation algorithms use global models, such as GMM [4] or Kernel Density Estimation [22], [23]. Moreover, global color and texture models discard the spatial information. A naive solution to this problem is obtained by using a spatial distance aware metric, such as *geodesic distance*. In



[19], spatio-temporal geodesic distance of each pixel to interaction points are computed and used in a graph-cut scenario. Although, this method significantly increases the number of frames that can be segmented without extra user interaction, geodesic distances are not robust against color profile changes or high amount of motion. Therefore, user interaction might be required in many frames which decrease the interaction quality significantly. On the other hand, in [18], energy function is defined over local overlapping windows. Therefore, locality of the energy function is significantly emphasized. These local windows are propagated by using the computed optical flow information. As a motion information-aware method, the main drawback of this method is its time complexity [18]. In summary, fully automatic video segmentation methods require computationally expensive computation of motion information for a successful operation. On the other hand, interactive methods exploit computationally efficient algorithms producing low quality video segmentation results and these methods try to overcome possible segmentation quality drawback via extensive and redundant amount of user interaction. However, extensive amount of user interaction reduces the quality of the user experience. When an interactive multimedia application is considered, efficiency, segmentation quality and quality of user interaction are all equally crucial. Therefore, the interaction should be at the minimum level, whereas the algorithm should be efficient and the results should be accurate. Indeed, these requirements pose a serious dilemma for video segmentation problems in multimedia applications. In this paper, in order to solve this dilemma, we redefine the video object segmentation problem as an estimation of the foreground probability density function of each frame in terms of probability density functions of previous frames. By the help of Markovian property assumption, we can define this problem as estimating the MRF energy of each frame in terms of MRF energy for the previous frame. In our framework, we assume that MRF-based energy function for the first frame is already known. In practice, this unknown energy function can be

obtained via user interaction or estimating the optical flow between the first two frames. We propose an efficient method to propagate energy distributions throughout the frames via bilateral filters. The weights of these filters are selected by using texture similarities and spatio-temporal relations. Moreover, an increase in the segmentation quality is expected by the help of simultaneous usage of spatial and texture cues. In order to increase the efficiency of the method further, we also propose a dynamic algorithm for the solution of the min-cut/max-flow problem in bilateral filtering scenario. The proposed method does not use any global shape or color model, it also does not use motion information. We compensate lack of motion information via extensive usage of spatial information during the calculation of bilateral filter weights. Indeed, for the videos having low amount of motion, the proposed method results in high quality segmentation even for non-rigid object motion. The proposed method is also general enough to convert any MRF-based interactive image segmentation algorithm to an interactive video segmentation method. Moreover, it can also be used to speed-up any automatic video segmentation tool. Our major contributions in this paper can be summarized as:

- 1) Utilization of bi-exponential filters to estimate the MRF energy of each frame from a consecutive previous frame that yields efficient and high quality video object segmentation;
- 2) Efficient and dynamic method to solve min-cut/max-flow in linear filtering scenario via linear recycling of residual flows. The proposed algorithm is explained in three main parts: In Section II, a general method to estimate the MRF energy of a frame by using MRF energy of previous frame is proposed via spatio-temporal bilateral filtering. In Section III, an efficient approximation of geodesic bilateral filter is introduced to increase the computational efficiency. Finally, in Section IV, a dynamic method to efficiently solve graph-cuts in linear filtering scenarios is proposed.



II. ESTIMATION OF MRF ENERGIES :

The proposed framework solves video segmentation problem for each frame separately in order to reduce the dimension since solving 2D graph-cuts is more efficient than solving 3D spatiotemporal graph-cuts. Hence, we can define video segmentation problem as estimating the MRF energy of the current frame by using the MRF energy of the previous frame. In other words, the proposed estimation methodology propagates the energy functions throughout the frames. Before starting to explain the details of this propagation method, we should summarize the usage scenario for this propagation. The proposed method can either be used as an interactive video segmentation tool or speed-up tool for an automatic video segmentation. Any MRF energy minimization-based interactive image segmentation method [4], [9], [6] can be used to generate MRF energy of the initial frame. Then, the obtained energy function can be propagated via the proposed algorithm. Hence, the overall algorithm can be used as an interactive segmentation tool. For an automatic video segmentation case, any optical flow based method exploiting MRF energies can be used to segment the initial frame [16]; then, the resultant energy can be propagated via the proposed method to speed-up the algorithm for the rest of the video. In order to explain the method which estimates MRF energies in terms of the MRF energy of the previous frame, we first need to state the explicit form of the energy function. As explained in [24], there exist a particular class of MRF energies which can be optimized efficiently via min-cut/max-flow method; namely submodular energy functions [34]. We are specifically interested in this class of energy functions. Generally, these energies composed of two main parts; the first part is the unary term which represents the consistency of the segmentation labels against some predefined models. The other term is the binary one and represents the coherency of the segmentation result. MRF energy over a graph, $G(\mathcal{V}, \mathcal{E})$, can be represented as

$$E(\alpha) = \sum_{v_i \in \mathcal{V}} U(\alpha_i, z_i) + \sum_{v_i \in \mathcal{V}} \sum_{v_j \in N(v_i)} V(z_i, z_j) \phi[\alpha_i \neq \alpha_j]$$

Although this formulation is usually applied to image data whose each pixel is a node, in order to increase the efficiency, over-segmenting an image is also a common technique [17], [9], [16]. Hence, we will refer to nodes as regions in the image; these regions might correspond to pixels or over-segments depending on the application. In this representation, represents the label of the region, represents the color, shape and/or motion information related to region and represents the neighbors of region. is an indicator function, yields one, if is true and 0 otherwise. is the unary energy corresponding to feature vector and label. Moreover, term represents the penalty associated for giving different labels to two neighbor regions, where represents the concatenation of labels of each region in the image. As a clarification to the notation, corresponds to set of nodes (set of) and corresponds to the binary energy terms associated with node and. When we consider the limited amount of camera, as well as object motion, we can assume that there exist a matching region (i.e. a region with a similar appearance) in the previous frame for each region in the current frame, except for the occlusion regions. Therefore, one can argue that MRF energy of the current frame can be approximately represented in terms of the MRF energy of the regions in the previous frame. In order to proceed further, one should assume that there exist a distance metric between each region in the current frame and the previous frame. This distance metric should correspond to the spatiotemporal distance and texture similarity, simultaneously. Among alternatives, geodesic distance is a suitable metric to be used in such setting. Geodesic distance can be considered as the minimum cost among the paths connecting two specified regions. In the proposed framework, this cost can be defined as the weighted sum of the color differences along the spatiotemporal path and length of the path. A detailed analysis of the geodesic distances and their usage scenarios in image/video segmentation problem can be

found in [19]. In the proposed method, an approximation to the geodesic distance is utilized; the details of this distance metric are explained in Section III. The representation of the energy function in terms of the previous energy function is defined separately for unary and binary terms. For the unary terms (U), the unary energy of each region in the current frame can be obtained by the weighted sum of the energy of each region in the previous frame. Moreover, these weights can be selected inversely proportional to these distances between regions. In other words, for any region in time t , unary term is written in terms of the weighted sum of the unary terms in time $t-1$ as

$$U^t(\alpha_i^t, z_i^t) = \frac{1}{\gamma_i^t} \sum_{v_j^{t-1} \in \mathcal{V}^{t-1}} U^{t-1}(\alpha_j^{t-1}, z_j^{t-1}) e^{-dis(z_i^t, z_j^{t-1})} \quad (1)$$

where superscripts represent the time instants and $dis(z_i^t, z_j^{t-1})$ represents the aforementioned distance metric between region in time and region in time $t-1$. γ_i^t is used for normalization and it can be computed as $\gamma_i^t = \sum_{v_j^{t-1} \in \mathcal{V}^{t-1}} e^{-dis(z_i^t, z_j^{t-1})}$. The main rationale behind the relation in (1) can be explained by considering each region in the previous frame as a model, and assuming the current setup as a mixture of these already calculated models. As in the case of mixture models, the resultant unary energy is a weighted sum of the each model in the mixture. Another rationale can be put forward by considering each region in the previous frame as a moving object. Therefore, every region in the current frame will either corresponds to an object in the previous frame or alpha matting of different overlapping objects. When the small segments generated by an over-segmentation algorithm or even pixels are considered as regions, every region in the current frame should have many matches in the previous frame; therefore, instead of exact matches, linear combination of all matches can be used. The selection of these matches and computation of their corresponding weights are implicitly performed by the help of the proposed distance metric. For the binary terms (V), the conventional approach is using

the frequently utilized Potts Model; (V is used for normalization). This penalty is generally considered as

inversely proportional to the color differences in order to generate coherent segmentation results. However, if we use this conventional approach, it is impossible to use the proposed dynamic optimization method explained in Section IV because both unary and binary energy terms of current frame are assumed to be linearly dependant on unary and binary energy terms of previous frame. Therefore, we slightly change the coherence penalty; we consider the graph of edges (i.e. dual graph of nodes) and apply the same propagation rule to this graph. Therefore, the relation between the current and previous binary energies are represented as follows:

$$V^t(z_i^t, z_j^t) = \frac{1}{\gamma_{ij}^t} \sum_{v_k \in \mathcal{V}^{t-1}} \sum_{v_l \in N(v_k)} e^{-dis(z_i^t, z_k^{t-1})} \cdot e^{-dis(z_j^t, z_l^{t-1})} V^{t-1}(z_k^{t-1}, z_l^{t-1})$$

(2) This definition of the binary energies can be related to smoothing the conventional binary terms. For the case of linear and piecewise linear penalty functions, one can show that the definition of the proposed binary penalty is equivalent to applying spatio-temporal edge-aware smoothing to video frames; then, computing conventional binary terms. In other words, defined binary energy is equivalent to the conventional binary energy used in [8] with an extra prior smoothing operation. Hence, defined binary energy can be considered as color smoothness penalty with an additional, temporal, smoothness.

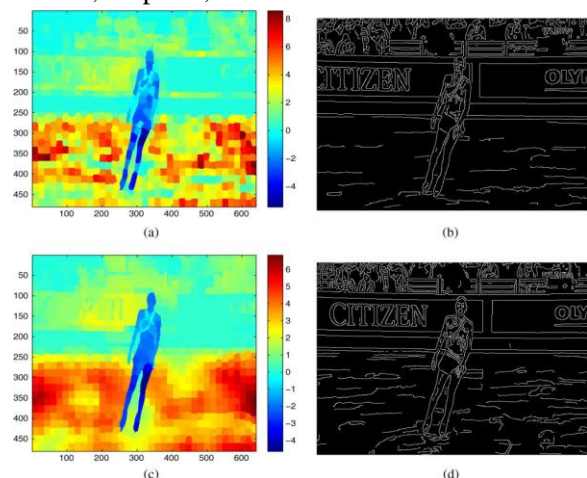


Fig. 1. Unary $U^t(\alpha_i^t, z_i^t)$ (a) and Binary energy $V^t(z_i^t, z_j^t)$ (b) terms of frame $t = 1$ is computed via user interaction, whereas energy terms of frame $t = 5$ is estimated via proposed energy propagation method. Estimated unary energy (c) is consistent with the actual foreground/background probabilities of the regions. Furthermore, estimated unary energy (c) is much smoother than the unary energy found via interaction (a) due to the wide support region.

Main advantages of the proposed estimation method are generality in terms of distance function selection and extended support region. Although the proposed system is based on geodesic distances, the proposed energy propagation can also be accomplished by a variety of distance functions accompanying color, motion and shape. Moreover, in this formulation, each region in the previous frame actually supports every region in the current frame. Therefore, any region is supported by all the regions of the previous frame. Such an approach relaxes limited motion assumption significantly, and eliminates the necessity to use any global model.

In order to visualize energy propagation, we apply the proposed method to a typical example. The energy of the first frame is computed via the interactive image segmentation method [9], and the energy of the 5th frame is estimated via the proposed method. As in the case of [9], over-segments are obtained by using SLIC algorithm [26]. The resultant propagation is illustrated in Fig. 1; where it suggests that the estimated unary terms are much smoother than the original terms due to the wide support region. It can also be observed that utilization of (2) for binary weights is feasible and the result is consistent with the edge map.

III. EFFICIENT COMPUTATION OF BILATERAL FILTERS - IP/BE FILTER

Energy propagation method explained in Section II, estimates the MRF energy of each frame. Hence, segmentation results can be obtained via minimization of these energies. However, the proposed energy propagation method is not applicable to interactive and mobile multimedia applications due to high computational complexity of bilateral filters and computation of geodesics. It should be noted that the relations in (1) and (2) correspond to a cross filtering

by a bilateral filter[32]. Naïve implementation of a bilateral filter requires operation; indeed, each operation requires a computation of a geodesic distance which has a linear time complexity via fast marching algorithm [27]. Therefore, the overall complexity of the estimation is which is far from being acceptable. In order to solve this efficiency problem, we utilize the recently proposed approximation of geodesic bilateral filter [32], [33]. Bilateral filters are widely used for edge-preserving smoothing in computer vision [28]. They can be considered as smoothing filters, whose weights are obtained from a Gaussian kernel. Later, bilateral filters are also attributed to anisotropic-diffusion filters [29]. However, their high computation cost make them unfavorable until some recent improvements over the computation time. In [30], bilateral filter is accelerated by using quantization and piecewise linear approximations. Alternatively, in [31], bilateral filtering is performed by using linear convolutions in higher dimensions. Recently, a constant time bilateral filter is proposed independently by Cigla *et al.* [32] and Thevenaz *et al.* [33] under different names as *Information Permeability (IP)* and *Bi-Exponential (BE) filter*. For the sake of fairness, we refer to this filter as IP/BE. In IP/BE filter, bilateral filtering is performed by using a dual one-tap recursive filter (IIR) for each dimension in the 2D spatial domain (vertical and horizontal). By the help of the recursive filter, one can achieve full image support for any pixel by a Gaussian kernel. 1D one tap recursion filter is applied in two directions (both horizontal and vertical) separately. Consider the sequence of length N , the recursion is applied as follows; For positive direction (left to right,[32],[33]):

$$\hat{x}_1[k] = x_1[k] + \hat{x}_1[k-1]r(x[k], x[k-1])$$

whereas, for negative direction (right to left):

$$\hat{x}_2[k] = x_2[k] + \hat{x}_2[k+1]r(x[k], x[k+1])$$

In this representation, r is the filter weight and can be computed as $r = \exp(-\frac{d^2}{2\sigma_d^2}) \exp(-\frac{I^2}{2\sigma_I^2})$. Moreover, this recursion should be initialized as $\hat{x}_1[0] = x_1[0]$. The values after recursion needs an extra normalization operation. During this

normalization, the constants are computed via the same recursion. Consider a sequence of 1's with length , as and , then the normalization constants are, for positive direction

$$\hat{1}_1[k] = 1_1[k] + \hat{1}_1[k-1]r(x[k], x[k-1])$$

and, for negative direction:

$$\hat{1}_2[k] = 1_2[k] + \hat{1}_2[k+1]r(x[k], x[k+1])$$

The final smoothed values are computed as [32], [33]:

$$y[k] = \frac{\hat{x}_1[k] + \hat{x}_2[k]}{\hat{1}_1[k] + \hat{1}_2[k]}$$

These 1D recursion filters can be applied to horizontal, spatial and temporal directions separately. Moreover, these results can be combined as explained in [32], [33]. It should be noted that, IP/BE filter is explained for smoothing scenario. In the proposed method, IP/BE filter is used as a cross filter. values corresponds to unary and binary energy terms at and, values correspond to unary and binary energy terms at .Moreover, the

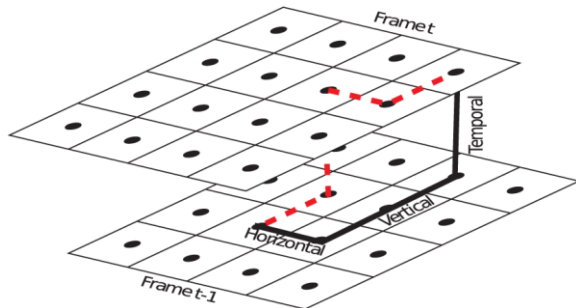


Fig. 2. Illustration of step-like paths used in IP/BE filter. Instead of a geodesic path shown as red dashed line (arbitrarily selected), a path having single horizontal, vertical and temporal components is used.

weight values are obtained via color values of the image instead of the energy terms. In other words, is replaced with . It should also be noted that these 1D filters are applied on horizontal, vertical and temporal dimensions separately. Hence, their ordering changes the result. We apply the filters in both orders in the spatial domain, then in the temporal domain (horizontal, vertical, temporal and vertical, horizontal,

temporal), and use the average of the results as the final propagated energy. In [32], it is argued that IP/BE filter approximates a geodesic bilateral filter. Although the weights do not correspond to the geodesic distances, they correspond to the cost of the path having a single horizontal and a single vertical piece. Moreover, the quality of this approximation is demonstrated in an edge preserving smoothing problem [32], [33]. Although this approximation could have some failure cases, it can still be used within (1) and (2) in most of the practical cases. When such filtering is used, the resulting MRF energy for the current frame is equivalent to the energy defined in Section II by as the cost of the step like path shown in Fig. 2. In other words, instead of taking the distance as the minimum of the sum of color differences among the paths connecting point and , this distance is taken as the sum of color differences along the path having three components on each dimension (and). For the videos not having serious level of noise or high amount of motion, the proposed cost approximates the geodesic distance [32].

When the proposed method is analysed, the number of operations is linear with the number of regions in the frames, if the number of overlapping regions is bounded by a constant. Indeed, most of the over-segmentation algorithms has a parameter which control the number of superpixels or the maximum size of a superpixel. Therefore, maximum number of overlapping regions can be adjusted. Hence, by proper selection of over-segmentation parameters, the proposed method runs in linear time. In summary; it is possible to approximate MRF energy propagation explained in (1) and (2) via algorithm having linear time complexity by using the IP/BE filter.

IV. BILATERAL DYNAMIC GRAPH-CUT

We propose a linear-time method to estimate MRF energy of a frame by using the previous frame. Moreover, for each frame, this energy function can be minimized via min-cut/max-flow method optimally [8]. However, due to our challenging efficiency requirements, we need to further improve min-cut/max-flow approach to increase the overall

efficiency. Hence, we propose a method to recycle residual flows in a bilateral filtering scenario for non-lattice graphs.

In general, MRF energy minimization problem is converted to min-cut problem on two terminal graph as an initial step. In the constructed graph, there are two terminals, namely source and sink (i.e. foreground and background) and every node is connected to these terminals by the terminal edge weights equal to unary energies. Moreover, non-terminal edge weights are selected as the binary energy terms. One can show that finding minimum cut on this graph is equivalent to minimizing the MRF represented energy. It is also shown that [35] finding minimum cuts is equivalent to finding maximum flows which can be pushed from source to sink. Typically, the solution to the max-flow problem is obtained via augment paths algorithm [8]. This algorithm can be explained by using the residual capacities and augmenting paths. Residual capacity of the edge is the maximum additional flow that can be sent from to through edge. The augmenting path is the path from source to sink through unsaturated residual edges (edges with positive residual weights). Augmenting paths algorithm uses the fact that pushing any flow through an augmenting path does not change the solution [8]. In other words, the solution to the original graph, and the graph which results from pushing a flow through an augmenting path is equivalent. The algorithm iterates until there exist no augmenting path in the graph; then, min-cut is the cut through edges having 0 residual weight. Since the cost of this cut is equal to zero, by sub-modularity, there can not be better solution. On the other hand, solving each frame separately is redundant. Augmenting flows in the previous frame can actually be "recycled" for the next frame. There exist a dynamic method [10] to solve graph-cut for each frame dynamically. In this dynamic algorithm [10], the structure of the graph assumed to be not changing throughout the video, and the edge weights change slightly throughout iterations. Therefore, if the residual graph of the previous iteration can be used, the computational burden might significantly decrease. In [10], a method for this edge update is developed. If a weight of edge is changed to,

than the solution to this new graph can be determined by solving the updated residual graph by the update. Indeed, this relation corresponds to pushing same flows through same nodes in the next frame. Only possible problem is negative edge weights and it is already solved in [10]. However, this method is not applicable to a general case, since the structure of the graph might change significantly due to superpixel-based representation. Over-segment positions and sizes might change in each iteration of the over-segmentation method. Therefore, we propose another dynamic version of the min-cut/max-flow algorithm for such a varying graph structure case in linear filtering scenario.

In order to apply the conventional dynamic method to varying graph structure case, a computationally expensive graph matching problem need to be solved between graph of the current frame and that of the previous frame. However, in the proposed method, there exist a linear relation between each node in the current frame and the previous frame. Moreover, edge weights of the graph of the current frame is defined in

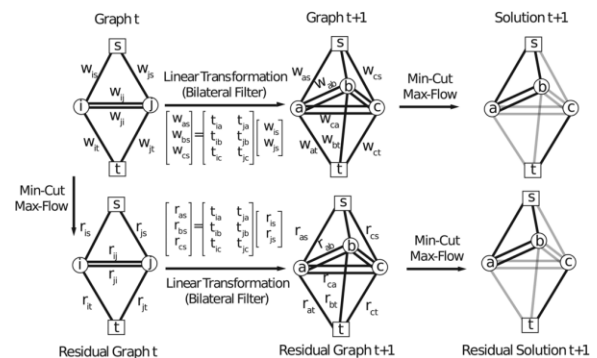


Fig. 3. Example for Proposition 1. The solution at t is obtained by applying linear transformation to the graph at t and solving graph at t . On the other hand, residual graph at t is obtained by applying same linear transformation to the residual graph at t which is already computed while solving graph at t . Moreover, solution for $t+1$ is obtained by solving residual graph at $t+1$. Proposition states that the solution at t and residual solution at t are equal. (In order to make the illustration less dense, and are not shown, in figure)



terms of the edge weights of the graph of the previous frame, as in (1) and (2). Therefore, this relation needs to be exploited. We propose to propagate the flows in the previous frame to the current frame via bilateral filter computed during the estimation step of MRF energies. More interestingly, we show that if the flows in the previous frame are propagated and pushed to the current frame, the resultant residual graph will be equivalent to applying the same bilateral filter to the residual graph of the previous frame. In other words, in order to find the updated residual graph in time, we simply apply the proposed bilateral filter to the residual graph in time. Moreover, we claim that minimum cut result will be same. This claim is also explained in Fig. 3 by an example. In order to keep the results as general as possible, we prove this claim for the general bilateral filtering scenario.

Proposition 1: Binary labels obtained by minimizing the MRF energy, resulted after applying bilateral filter on the energy function which is defined via residual graph, is equivalent to minimizing the MRF energy obtained via applying bilateral filter on the original energy function.

The proof of this proposition is deferred to Appendix. By using Proposition 1, instead of propagating MRF energy completely, we only propagate residual graphs throughout the frames. This dynamic propagation results in significant computation time decrease as explained in Section V-B. Efficiency obtained via dynamic graph-cut can be crucial factor to increase the interaction quality for interactive multimedia applications. On the other hand, the proposed technique is the only dynamic min-cut/max-flow method which is applicable to the graphs whose structures are varying. In summary, the proposed video segmentation method starts with an already known MRF energy for the first frame, solves the min-cut/max-flow problem; then, propagates the residual energy to the next frame via the proposed linear time algorithm and continues to iterate through frames.

V. EXPERIMENTAL RESULTS

As explained in Section I, the proposed MRF energy propagation can either be used as an interactive video segmentation algorithm with an existing interactive image segmentation method or as a speed-up method for an automatic video segmentation algorithm. Hence, we have experimented the proposed method for both scenarios. In Section V-A, we utilize recently proposed mobile and dynamic image segmentation tool [9] in order to experiment the interactive video segmentation scenario. Moreover, segmentation quality and computational efficiency results are compared against existing interactive video segmentation algorithms. On the other hand, in Section V-B, we use the proposed method to speed-up a recently proposed automatic video segmentation tool [15]. We compare segmentation quality and computational efficiency of the original method and proposed speed-up extension. Within the experimental procedures, we tried to keep effect of parameters as small as possible. Indeed, only set of parameters which can be tuned are related to either SLIC over-segmentation [26] or IP/BE Filter. We kept the default parameters of SLIC over-segmentation, and chose 1000 as the number of over-segments. Kernel size of the IP/BE filter is yet another parameter to be chosen. We chose 30 as the kernel size for, and kernels. Discussion related to effect of kernel size can be found in [32].

A. Interactive Video Segmentation :

For the evaluation of the interactive video segmentation scenario, we utilize "coloring-based" interactive image segmentation technique [9] to segment the initial frame. Then, we use the proposed method to propagate the initial segmentation results. Throughout the video, only the initial frame is interacted; no other user interaction is applied for the other frames. We compare the proposed method against two high-performing interactive video segmentation methods from the literature [22], [18] and one tracking based method [37]. The first one is a geodesic segmentation method [22]. Due to the fact

that permeability filter [32] actually approximate geodesic distances, our proposed method approximates a bilateral filter using geodesic distances as the filter weights. Therefore, our algorithm can be accepted as an extension of [22] with a higher support region, additional coherency term and smoother energy propagation. The other algorithm is a local classifier based segmentation method [18] which is included in Adobe After Effects CS5 [36] as the *roto-brush tool*. Motion coherent tracking is one of the state-of-the-art tracking based segmentation methods. We use exactly same interaction for the segmentation of initial frames for all algorithms. Then, without any further interaction, we use these algorithms for entire video sequence.

1) Quality: For the subjective comparison of the algorithms, we have used the dataset used in [16]. For the *iceSkater* sequence, the resultant segmentation and input video is presented in Fig. 4. For the subjective comparison, only interactive video object segmentation methods are included. It can be observed in Fig. 4, *roto brush tool* [18] and the proposed method have similar performances. On the other hand, geodesic video matting tool [22] fails to propagate the segmentation results. The main reason for this performance drawback is its single pixel support region. The motion information is also discarded in the geodesic video matting tool. For the first 50 frames, both the proposed method and *roto brush* [18] gives near optimal segmentation results.

Then, both algorithms segments part of the background as foreground erroneously. In addition to these, at frames, 76 and

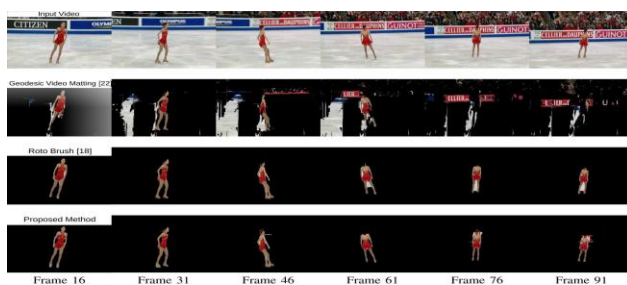


Fig. 4. Visual comparison of interactive video segmentation algorithms for *iceSkater* sequence. Top row shows the original video, second row shows the result of *geodesic video segmentation* tool [22], third row shows the result of *rotobrush* tool [18], and bottom row shows the result of proposed algorithm. Superior performance of the *rotobrush* and proposed method against *geodesic video segmentation* is clear in all frames. Superior performance of the proposed method against *rotobrush* can also be observed in Frame 61 and 76.

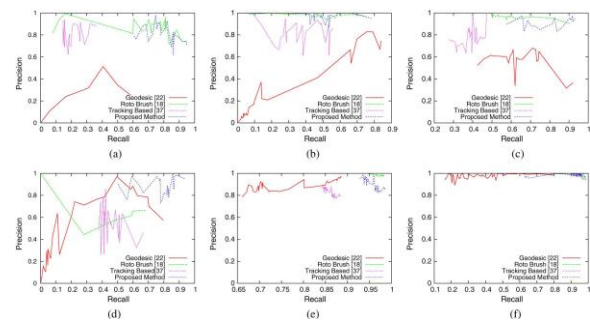


Fig. 5. Precision-Recall curves for SegTrack [37] dataset (A curve near to top-right corner indicates better performance). Precision-recall curves suggest that for interactive video segmentation problem, the proposed method shows either superior or comparable performances against the available methods in the literature. (a) Birdfall. (b) Cheetah. (c) Girl. (d) Monkey. (e) Penguin. (f) Parachute 91, the region around legs of the ice skater is segmented as foreground by *roto brush*. The main reason for this is the shrinking bias of MRF energies [8]. However, the proposed method handles shrinking bias successfully due to its wide support region. Hence, one can conclude that the segmentation quality of the proposed method is slightly superior than the *roto brush* [18]. For the quantitative comparison of the algorithms, we have used *Segtrack* dataset [37]. The initial frame is segmented by using the same user interaction, and the algorithms are executed for the rest of the video. Then, pixel-based precision and recall values are computed for each frame via the following relations and ; where is true positive, is false positive and is false negative. The resulting precision recall curves for each video sequence is plotted in Fig. 5. Due to the space constraints, visual comparisons are not included in the paper. However, they can be reached through



Fig. 6. Visualization of the motion blur and color characteristic in SegTrack [37] dataset. Motion blur in the *Girl* sequence is clearly visible in (a). The black monkey in *Monkey* sequence, is at the image boundary; therefore, local windows have no color information. Moreover; in *penguin* dataset, ground truth is selected as a single penguin. Hence, interactive image segmentation algorithm fails due to the repetitive structure of the image. (a) Girl Frame 16. (b) Monkey Frame 19. (c) Penguin Frame 1.

the web-page of the paper (<http://www.ozansener.net/mrfpropagation/>). It can be observed from Fig. 5 that in all videos, both the proposed method and *rotobrush* [18] have superior performance against *geodesic method* [22]. The main reason for this result is the incorporation of motion information in [18] and implicit usage of spatial information and wide support region (whole frame) for the proposed method. This result clearly indicates the necessity of the spatial information usage in video segmentation problem. On the other hand, *tracking based method* have competitive precision values and slightly worse recall values. Moreover, proposed method over performs *tracking based method* in all videos. For *BirdFall* sequence, the proposed algorithms has better recall values then *rotobrush* [18]. For the points having recall value higher than 0.6, the algorithms show almost same performance. For the *Cheetah* sequence, *rotobrush* and the proposed method have again similar performances. For the *Girl* sequence, *rotobrush* [18] outperforms the proposed method. This

behaviour is due to the high motion blur in the sequence. For the motion blur case, color differences are low even at the object boundaries; therefore, the filter coefficients are not computed properly. An example of the motion blur in the dataset is shown in Fig. 6. Hence, in *Girl* sequence, there is almost no edge information to be used in the frames due to themotion blur; therefore, the proposed method fails.

For *Monkey* sequence, the proposed method out performs *rotobrush* [18]. The motion occurs around the frame boundary; therefore, there is not enough local color information to be used for *rotobrush* [18]. This characteristic of the video is also visualized in Fig. 6. For *Penguin* sequence, utilized interactive image segmentation algorithm fails to segment the initial frame. Hence, the performance drawback is not due to the proposed energy propagation method. An example image is also shown in Fig. 6. Finally, for the *Parachute* sequence, both *rotobrush* [18] and the proposed method have precision and recall values higher than 0.9; both algorithms have similar precision values; however, the proposed method have better recall values.

2) *Computation Time*: In order to compare the computational efficiencies of the algorithms, we have used the *SegTrack* [37] dataset. Each sequence in the dataset is rescaled to 640x480 resolution, and, segmentation time for each frame is computed for all sequences. The mean computation time for each frame is summarized in Table I.

As it can be observed from the Table I, the proposed method has the highest computational efficiency. It should also be noted

TABLE I
OVERALL COMPUTATION TIME PER FRAME

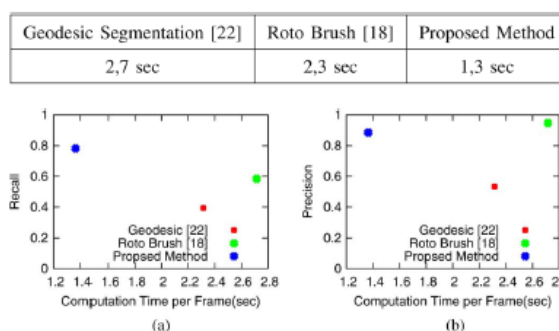


Fig. 7. Recall and Precision versus Computation Time. The results suggest that the proposed method reaches twice the efficiency with superior recall and comparable precision values. (a) Recall vs Computation Time. (b) Precision vs Computation Time.

that, the significant part of the computation time for the proposed method is consumed by SLIC [26] algorithm. Convergence of SLIC [26] algorithm takes approximately 0.9 second per frame. Hence, by the introduction of efficient over-segmentation algorithms, it is possible to reach much better computation time for the proposed method. In order to visualize the performance vs time trade-off for all the algorithms, we have also performed performance vs time comparisons. For the *SegTrack* [37] dataset, we have computed mean precision and mean recall values as well as the corresponding computation times. Precision vs computation time and recall vs computation times are presented in Fig. 7. As it can be observed from Fig. 7(a), the proposed method has better recall and computation time, when it is compared against other algorithms. In Fig. 7(b), *rotobrush* [18] has slightly better precision values with a higher computational burden. One can conclude that the proposed method reaches segmentation quality of *rotobrush* [18] with much higher computational efficiency. On the other hand, *tracking based method* [37] is not included in this plots since its computation time per frame is around 1 minute. Moreover, computation time of the *tracking based method* is not comparable with other algorithms. Indeed, its computational complexity is not affordable in any interactive system. 3) *Number of Interacted Frames*: In order to analyze the effect of the number of interacted frames, we designed another

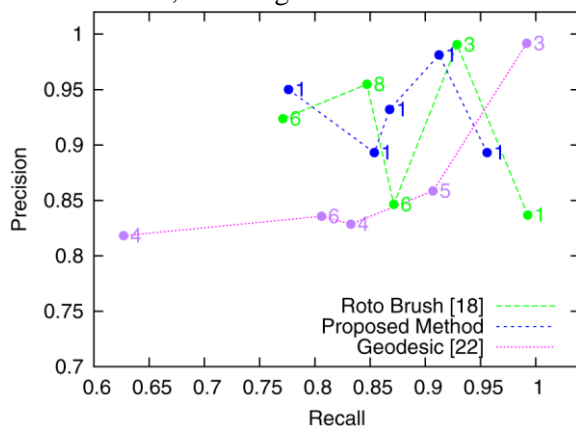


Fig. 8. Number of interacted frames, precision and recall values for each sequence in SegTrack [37] dataset. Each point represents a video sequence;

precision and recall values are y and x coordinates respectively and the number of interacted frames are shown right next to the data points. First 30 frames of each sequence is used for the experiment; hence, number of frames in each sequence is equal.

experiment. Since the proposed algorithm is designed for an interaction in the initial frame, only the first frame is used for the interaction for the proposed method. Whereas; for the experiments on *roto brush* [18] and *geodesic method* [22], we let user interact on as many frames as he/she wants. User initially interacts with the first frame and see the segmentation result for the rest of the frames; then, user updates the segmentation by choosing another key-frame and giving interaction for it and so on. Within the experiments, SegTrack [37] data set is used. For each video, precision, recall as well as number of interacted frames is recorded. In Fig. 8, precision-recall curve over SegTrack [37] dataset is shown. Precision-recall curve is augmented with the number of interacted frames for each sequence. As shown in the Fig. 8, proposed method and *roto-brush* [18] have similar precision and recall values. On the other hand, *roto-brush* requires interaction on 3 to 8 frames for comparable performance with the proposed method which only requires interaction on the first frame. Moreover, *geodesic method* [22] can not reach to the performance level of the proposed method and *roto-brush* even with high number of interactions.

B. Dynamic Bilateral Graph-Cut

In order to experiment the computational improvements obtained by dynamic computation of graph-cut in filtering scenario, we have performed MRF energy propagation via both conventional min-cut/max-flow method [8] and the proposed dynamic-graph cut. As explained in Section IV, dynamic graph-cut presented in [10] is not applicable to our scenario, since the structure of the graph in the proposed method changes significantly due to the over-segmentation. For the min-cut/max-flow, we propagate MRF energy and applied min-cut/max-flow [8]. For the dynamic graph-cut, we use the proposed dynamic method. Only

the graph-cut execution times for each frame is plotted. As shown in Fig. 9, the proposed improvements results in a significant decrease in the computation time.

C. Automatic Video Segmentation Speed-up

As explained in Section I and II, the proposed energy propagation= tool can actually be used in order to speed-up any auto-

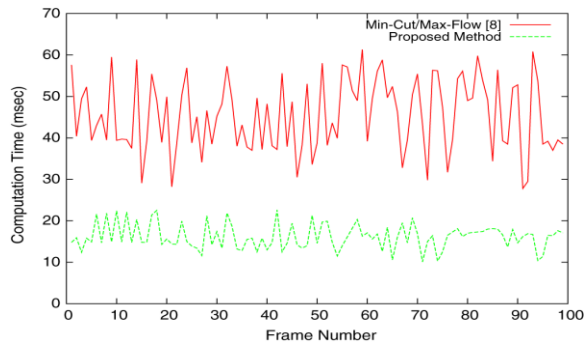


Fig. 9. Computation time for dynamic graph-cut of each frame. The proposed improvements results in around 3 times efficiency increase. matic video segmentation tool. We have experimented this scenario on a recently proposed automatic segmentation algorithm Key-Segments [15]. In Key-Segments algorithm, graph-based 2D clustering of frames are performed to generate as many hypothesis as possible. Then, these hypothesizes are ranked by using a saliency-like measure and this measure is computed by using both shape, texture and motion information. Both extraction of optical flow vectors and computation of saliency-like measure is computationally expensive. In our setup, Key-Segments [15] method is applied only to initial frames (). Then, the color information of both foreground and background is obtained via EM procedure in terms of Gaussian Mixture Models (GMM). Then, the energy function is defined as in the case of interactive image segmentation [9] only for initial frames. No infinite terminal weight is used due to the lack of definite user interaction. After the definition of the energy function for the first few frames, the proposed energy propagation

and minimization tool is used to propagate and solve the segmentation problem for the rest of the video. In order to analyze the performance of these speed-up, we have used *SegTrack* [37] dataset. As explained in [15], *Penguin* dataset is discarded due to the ambiguous ground truth. For the initial 3 frames, Key-Segments algorithm is performed and for the rest of the video, the proposed algorithm is used. The computation time and precision-recall values for each frame of each video is recorded and the resulting precision-recall curves are plotted in Fig. 10. The average computation time for each frame is also computed in summarized in Table II. For a fair comparison, source code distributed with [15] is used, and the proposed method is reimplemented in MATLAB. When the precision recall curves in Fig. 6 is considered; except for the *Girl* sequence, the proposed speed-up method performs comparable against the original Key-Segments [15] method. In *Birdfall* sequence, the proposed method has better recall values than the original Key-Segments [15]. Moreover, in *Cheetah* and *Monkey* sequences, both algorithms perform very similar to each other. In *Parachute* sequence, both algorithms performs near optimal and most of the performance drawback of proposed method is due to the superpixel errors caused by SLIC [26] algorithm. Finally, in *Girl* sequence the proposed algorithms perform much worse than the original Key-Segments [15].

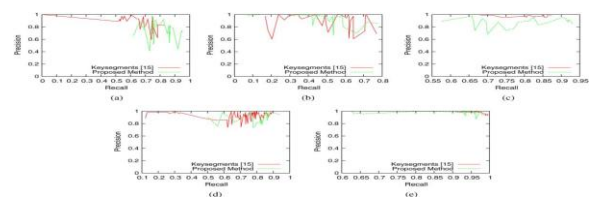


Fig. 10. Precision-recall curves for automatic video segmentation on SegTrack [37] dataset. The proposed method does not cause significant performance drawback in any of the videos, except *Girl* sequence. The reason for this performance drawback is motion blur. (a) Birdfall. (b) Cheetah. (c) Girl. (d) Monkey. (e) Parachute

TABLE II



COMPUTATION TIME PER FRAME (IN MATLAB)

Key-Segments [15]	Speed-up Key-Segments (Proposed Method)
260.6 sec	4.0 sec

The main reason of this performance drawback is due to the high amount of motion blur in the sequence. The edge information is used extensively by permeability filter; therefore, lack of edges results in serious performance drawback. Motion blur in the *Girl* sequence is visualized in Fig. 6, and discussed in Section V-A.I In conclusion, except for the case of very high motion blur, the proposed speed-up results in similar segmentation quality, when compared to the original segmentation algorithm. On the other hand, when computational complexity is considered, the proposed algorithm results in around 65 times speed-up when compared with the original algorithm. It is reasonable because main bottleneck of the Key-Segments [15] algorithm is its initial object proposal step and it is avoided in the proposed method.

VI. CONCLUSION :

In this paper, we address the problem of video segmentation from the perspective of computational efficiency. We redefine the problem as propagation of MRF energies through frames. We propose a novel technique via bilateral filters in order to increase computational efficiency. We further increase the efficiency by a novel dynamic graph-cut formulation applicable to a linear filtering scenario. The proposed method can either be used as an interactive video segmentation tool or a speed-up tool for automatic video segmentation. The experimental results suggest that in both scenarios, the proposed methods produce segmentations having higher or comparable quality when compared against the leading algorithms from the literature. From the perspective of efficiency, the proposed method results in 2 times increase in the computational efficiency for interactive video segmentation and 65 times increase in the computational efficiency for automatic video

segmentation with almost no significant decrease in segmentation quality.

APPENDIX A PROOF OF THE PROPOSITION 1

Proposition 1: Binary labels obtained by minimizing the MRF energy, resulted after applying bilateral filter on the energy function which is defined via residual graph, is equivalent to minimizing the MRF energy obtained via applying bilateral filter on the original energy function.

Proof : We prove the proposition by constructing flows at time t . The flows are constructed by applying bilateral filter to the flows at time $t-1$. We show that these flows converts the propagated energy function to a propagated residual energy. We will consider the flows in terminal and non-terminal weights separately. Furthermore, our proof will be based on edge flows; not source-sink flows. Source-sink flows can be obtained via concatenation of edge flows. For the flows on terminal edges, consider (v_i is either source or sink) as flows in time $t-1$.

$$f_{jT}^t = \frac{1}{\gamma_i^t} \sum_{v_i \in V^{t-1}} f_{iT}^{t-1} e^{-dis(z_i^{t-1}, z_j^t)} \quad (3)$$

For the non-terminal flows, consider f_{ij}^{t-1} as flows in $t-1$

$$f_{ij}^t = \frac{1}{\gamma_{ij}^t} \sum_{v_k \in V^{t-1}} \sum_{v_l \in N(v_k)} f_{kl}^{t-1} e^{-dis(z_k^{t-1}, z_i^{t-1})} e^{-dis(z_j^t, z_l^{t-1})} \quad (4)$$

It is shown that applying any flow through graph does not change the solution to minimum cut problem [8]. Therefore, the solution after applying these flows does not change. The residual weights are obtained after applying these flows as

$$\begin{aligned} r_{iT}^t &= U^t(t, z_i^t) - f_{iT}^t \\ &= \frac{1}{\gamma_i^t} \sum_{v_j^{t-1} \in V^{t-1}} U^{t-1}(t, z_j^{t-1}) e^{-dis(z_i^t, z_j^{t-1})} \\ &\quad - \frac{1}{\gamma_i^t} \sum_{v_j \in V^{t-1}} f_{jT}^{t-1} e^{-dis(z_i^t, z_j^{t-1})} \\ &= \frac{1}{\gamma_i^t} \sum_{v_j^{t-1} \in V^{t-1}} r_{jT}^{t-1} e^{-dis(z_i^t, z_j^{t-1})} \end{aligned} \quad (5)$$



A similar relation ($r_{ij}^t = V^t(z_i^t, z_j^t) - f_{ij}^t$) can also be obtained for the non-terminal weights as well

$$\begin{aligned} r_{ij}^t &= \frac{1}{\gamma_{ij}^t} \sum_{v_k \in V} \sum_{v_l \in N(v_k)} e^{-d(z_i^t, z_k^{t-1})} e^{-d(z_j^t, z_l^{t-1})} V(z_k^{t-1}, z_l^{t-1}) \\ &\quad - \frac{1}{\gamma_{ij}^t} \sum_{v_k \in V^{t-1}} \sum_{v_l \in N(v_k)} f_{kl}^{t-1} e^{-dis(z_i^t, z_k^{t-1})} e^{-dis(z_j^t, z_l^{t-1})} \\ &= \frac{1}{\gamma_{ij}^t} \sum_{v_k \in V} \sum_{v_l \in N(v_k)} e^{-d(z_i^t, z_k^{t-1})} e^{-d(z_j^t, z_l^{t-1})} r_{kl}^{t-1} \quad (6) \end{aligned}$$

This result corresponds to applying bilateral filter to the weights in the residual graph. In summary, we prove that propagating flows used in the min-cut/max-flow and recycling them for the next frame, corresponds to propagating residual graph via bilateral filters.

REFERENCES:

- [1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Pattern Anal. Mach. Intell.*, vol. 22, pp. 888–905, 1997.
- [2] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, pp. 167–181, Sep. 2004.
- [3] L. Grady and E. L. Schwartz, "Isoperimetric graph partitioning for image segmentation," *IEEE Pattern Anal. Mach. Intell.*, vol. 28, pp. 469–475, 2006.
- [4] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *Proc. SIGGRAPH 2004*.
- [5] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Trans. Graph.*, vol. 23, p. 303, Aug. 2004.
- [6] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. ICCV 2001*.
- [7] J. Liu, J. Sun, and H.-Y. Shum, "Paint selection," in *Proc. ACM SIGGRAPH 2009*, New York, NY, USA, 2009, pp. 69:1–69:7.
- [8] Y. Boykov and V. Kolmogorov, "An experimental comparison of mincut/ max-flow algorithms for energy minimization in vision," *IEEE Pattern Anal. Mach. Intell.*, vol. 26, pp. 1124–37, Sep. 2004.
- [9] O. Şener, K. Uğur, and A. A. Alatan, "Error-tolerant interactive image segmentation using dynamic and iterated graph-cuts," in *Proc. 2nd ACM Int. Workshop Interactive Multimedia on Mobile and Portable Devices (IMMPD)*, 2012.
- [10] P. Kohli and P. H. S. Torr, "Dynamic graph cuts for efficient inference in Markov random fields," *IEEE Pattern Anal. Mach. Intell.*, vol. 29, pp. 2079–2088, 2007.
- [11] H. Lombaert, Y. Sun, L. Grady, and C. Xu, "A multilevel banded graph cuts method for fast image segmentation," in *Proc. ICCV'05*, pp. 259–265.
- [12] A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image sequence analysis for emerging interactive multimedia services-the European COST 211 framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 802–813, Nov. 1998.
- [13] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *Proc. IEEE CVPR 2010*, Jun. 2010, pp. 73–80.
- [14] J. Kim and K. Grauman, "Boundary preserving dense local regions," in *Proc. 2011 IEEE CVPR 2011*, pp. 1553–1560.



- [15] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *Proc. IEEE ICCV*, 2011, pp. 1995–2002.
- [16] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph based video segmentation," in *Proc. IEEE CVPR*, 2010.
- [17] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, *Multiple hypothesis video segmentation from superpixel flows. ECCV'10*. Berlin, Germany: Springer-Verlag, 2010, pp. 268–281.
- [18] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video snapcut: Robust video object cutout using localized classifiers," in *Proc. SIGGRAPH' 09*, pp. 70:1–11.
- [19] A. Criminisi, T. Sharp, and A. Blake, "Geos: Geodesic image segmentation," in *Proc. ECCV '08*, 2008, pp. 99–112, Springer-Verlag.
- [20] Y. Li, J. Sun, and H.-Y. Shum, "Video object cut and paste," in *ACM SIGGRAPH 2005 Papers, SIGGRAPH '05*, 2005, pp. 595–600.
- [21] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen, "Interactive video cutout," *ACM Trans. Graph.*, pp. 585–594, Jul. 2005.
- [22] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *Proc. ICCV 2007*, 2007, pp. 1–8.
- [23] A. Elgammal, R. Duraiswami, and L. S. Davis, "Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking," *IEEE Pattern Anal. Mach. Intell.*, vol. 25, pp. 1499–1504, 2003.
- [24] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *IEEE Pattern Anal. Mach. Intell.*, vol. 26, pp. 147–59, Mar. 2004.
- [25] S. Geman and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. San Francisco, CA, USA: Morgan Kaufmann, 1990, pp. 452–472.
- [26] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, SLIC Superpixels tech. rep., EPFL, 2010.
- [27] L. Yatziv, A. Bartschaggi, and G. Sapiro, "O(n) implementation of the fast marching algorithm," *J. Computat. Phys.*, vol. 212.
- [28] V. Aurich and J. Weule, "Non-linear Gaussian filters performing edge preserving diffusion," in *Proc. DAGM-Symp.*, 1995, Springer.
- [29] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Pattern Anal. Mach. Intell.*, vol. 12, pp. 629–639, 1990.
- [30] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Proc. ACM SIGGRAPH '02*, 2002, pp. 257–266.
- [31] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *Int. J. Comput. Vision*, pp. 24–52, Jan. 2009.



[32] C. Cigla and A. A. Alatan, “Efficient edge-preserving stereo matching,” in *Proc. IEEE ICCV Workshops*, 2011, pp. 696–699.

[33] P. Thévenaz, D. Sage, and M. Unser, “Bi-exponential edge-preserving smoother,” *IEEE Trans. Image Process.*, vol. 21, pp. 3924–3936, 2012.

[34] V. Kolmogorov and R. Zabih, “What energy functions can be minimized via graph cuts?,” *IEEE Pattern Anal. Mach. Intell.*, vol. 26, pp. 147–159, Mar. 2004.

[35] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks*. Princeton, NJ, USA: Princeton Univ. Press, 1962.

[36] A. Systems, “Creative suite 5 user guide,” 2010.

[37] D. Tsai, M. Flagg, and J. M. Rehg, “Motion coherent tracking with multi-labelMRF optimization,” in *Proc. British Machine Vision Conf.*, 2010.

